

PATENT ABSTRACTS OF JAPAN

W1258

(11)Publication number : 2001-166956

(43)Date of publication of application : 22.06.2001

(51)Int.Cl.

G06F 9/46
G06F 15/00
G06F 15/177

(21)Application number : 11-345745

(71)Applicant : HITACHI LTD

(22)Date of filing : 06.12.1999

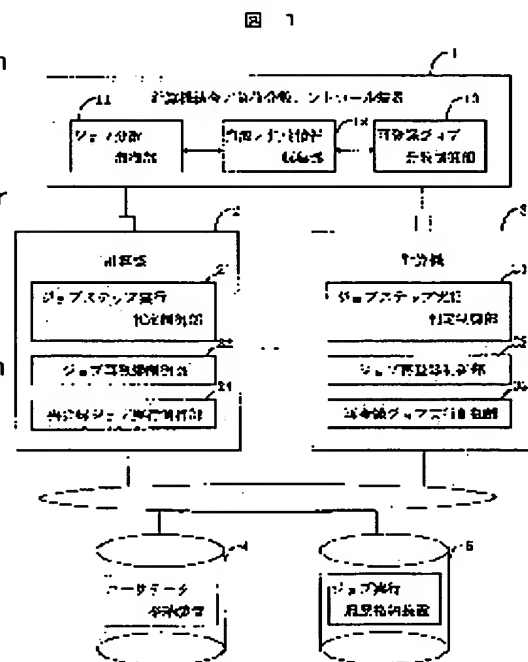
(72)Inventor : YAMAZAKI HIROKI
MATSUO SHIGEKI
KIHARA HITOSHI
ICHIKAWA MASATOSHI
OCHI NAOKO

(54) JOB SCHEDULING SYSTEM IN COMPOSITE SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To attain the well-balanced load distribution of job steps when one job is composed of plural job steps in a general computer system constituted of plural computers for preventing the execution delay or execution invalidity of the job step following the job to be executed due to the generation of the shortage of resources under the influence of the other job which is being executed by the computer during the execution of the job step.

SOLUTION: When a job to be executed is composed of plural job steps, at the time of executing the following job step, the resource use information of computers 2...3 is obtained by a resource/load information collecting part 13, and the optimal computer is selected, and the job is re-registered in the computer, and the interrupted job step is resumed.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-166956

(P2001-166956A)

(43) 公開日 平成13年6月22日 (2001.6.22)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)
G 0 6 F 9/46	3 6 0	G 0 6 F 9/46	3 6 0 C 5 B 0 4 5
15/00	3 1 0	15/00	3 1 0 H 5 B 0 8 5
15/177	6 7 4	15/177	6 7 4 A 5 B 0 9 8

審査請求 未請求 請求項の数 4 O L (全 6 頁)

(21) 出願番号 特願平11-345745

(22) 出願日 平成11年12月6日 (1999.12.6)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 山崎 弘樹

神奈川県横浜市戸塚区戸塚町5030番地 株

式会社日立製作所ソフトウェア事業部内

(72) 発明者 松尾 繁喜

神奈川県横浜市戸塚区戸塚町5030番地 株

式会社日立製作所ソフトウェア事業部内

(74) 代理人 100075096

弁理士 作田 康夫

最終頁に続く

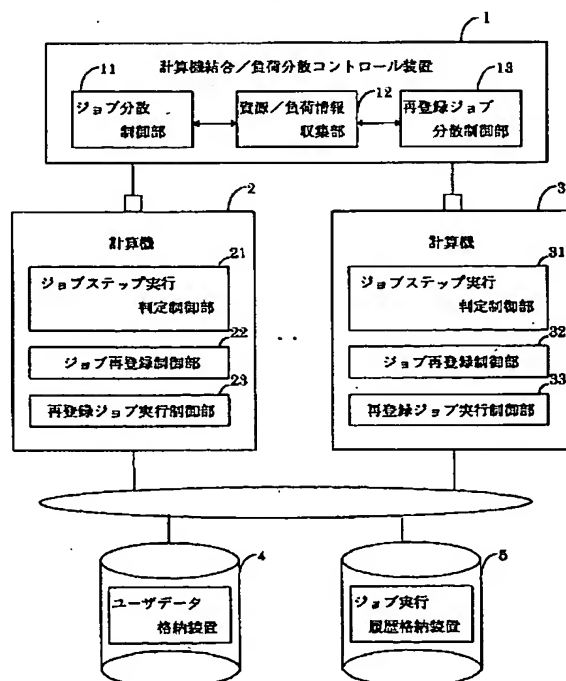
(54) 【発明の名称】 複合システムにおけるジョブスケジューリング方式

(57) 【要約】

【課題】 複数の計算機から構成される汎用計算機システムにおいて、1つのジョブが複数のジョブステップから構成されていた場合、ジョブステップ実行中、その計算機で実行している他のジョブの影響で資源不足等が発生すると、実行しようとする当該ジョブの後続のジョブステップが実行遅延、または実行不可となってしまうためにジョブステップをバランスよく負荷分散する必要がある。

【解決手段】 実行しようとするジョブが複数のジョブステップから構成されている場合、後続のジョブステップを実行する時、資源/負荷情報収集部13より計算機2・・3の資源使用情報を取得して、最適な計算機を選択後、その計算機へジョブを再登録し、中断されたジョブステップから続行する。

図 1



【特許請求の範囲】

【請求項1】 複数の計算機から構成される汎用計算機システムでのジョブスケジューリングにおいて、複数のジョブステップ構成からなるジョブを実行する場合、後続ジョブステップの実行を開始させる直前に、引き続き自計算機で実行させるかどうかの判断を、接続している全計算機の資源／負荷情報をもとに行うことを特徴とするジョブスケジューリング方式。

【請求項2】 ジョブステップの実行が引き続き自計算機で実行不可能と判断した場合、計算機結合／負荷分散コントロール装置の再登録ジョブ分散制御部へジョブの再登録要求を行うことを特徴とするジョブスケジューリング方式。

【請求項3】 計算機結合／負荷分散コントロール装置の再登録ジョブ分散制御部へジョブが再登録された場合、最適な計算機へジョブの再スケジュールすることを特徴とするジョブスケジューリング方式。

【請求項4】 計算機結合／負荷分散コントロール装置の再登録ジョブ分散制御部から計算機へジョブが再登録された場合、ジョブ実行履歴格納装置に格納してある履歴データをもとに当該ジョブが以前実行開始しようとしていたジョブステップから再実行を開始させることを特徴とするジョブスケジューリング方式。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】 本発明は、複数の計算機から構成される汎用計算機システムにおいて、実行させるバッチジョブの負荷分散ジョブスケジューリング方式に関する。

【0002】

【従来の技術】 複数の計算機から構成される汎用計算機システムにおいて、接続されたシステム上で業務処理を効率よく遂行するために各計算機の資源を有効に使用し、1つの計算機に負荷が偏らないような分散のためのジョブスケジュール方法は従来から研究されている。たとえば本発明に近い文献として特開平9-120389号で示されるものがある。それによればジョブ単位での負荷分散スケジュールを行うもので、ジョブ入力時、接続されている各計算機の資源の使用度合いを測定した結果から最適な計算機を選択し、その計算機にジョブを実行させようとするものである。何故、ジョブ単位のスケジュールかというと、通常、ジョブを計算機で実行するためには、専用のイニシエータという1仮想空間を使用して処理するが、そこでジョブの実行が開始されると、そのジョブの構成が1つのジョブステップから構成されていても複数のジョブステップから構成されていても、そのイニシエータ空間だけを使用して第1ステップから最終ステップまでを実行することになるからである。よって、ジョブと言う単位でしかジョブのスケジュールを行えない。

【0003】 また、上記で示す複数のジョブステップから構成されているジョブを、各ステップ毎に複数の計算機で並列に実行させるための手段も研究されている。その文献として特開平8-286932で示されるものがある。この機能は1つのジョブが複数のジョブステップで構成されている場合、ステップそれぞれを単独の1ジョブとして各計算機で処理するため、ジョブ全体のターンアラウンドタイムの短縮、システム全体のスループットの向上を図ることができる。しかし、実行させるステップ間の依存関係がない場合のみジョブステップを並列に各計算機で実行できるものである。

【0004】

【発明が解決しようとする課題】 上記で挙げたような従来技術では、ジョブ単位でのスケジュールは円滑に行っていた。しかし、1つのジョブが複数のジョブステップから構成されていた場合、選択された計算機でジョブが一旦実行されると、その後の計算機の資源使用の負荷状況にかかわらず、ジョブ全体がその計算機のみで実行される。ここで言う負荷とはCPU使用率、主記憶や補助記憶の使用率、I/O負荷率などがあるが、主にCPU使用率のことである。そのため、その計算機で実行している他のジョブの影響で資源不足等が発生すると、後続ジョブステップの実行遅延でジョブ全体のターンアラウンドタイムが延びたり、またはジョブステップが実行不可能となってしまう。

【0005】

【課題を解決するための手段】 本発明は複数の計算機から構成される汎用計算機システムにおいて、実行しようとしたジョブが複数のジョブステップから構成されおり、かつステップ間で依存関係がある場合、ステップ実行終了時に今実行している計算機で後続のジョブステップを実行するかどうかを判断し、実行すべきでないと判断したとき、実行を中断し、他の計算機に委ねるものである。まず、第1ステップのジョブステップが実行され終了すると、後続のジョブステップを実行しようとする。この時、計算機結合／負荷分散コントロール装置の資源／負荷情報収集部より各計算機の資源使用情報を得て、自計算機の資源使用の負荷が高く実行できないと判断した場合、一旦、ジョブステップの実行を中断し、計算機結合／負荷分散コントロール装置の再登録ジョブ分散制御部へジョブの再登録要求を行い、再登録ジョブステップ分散制御部より最適な他の計算機へジョブを転送し、再投入された計算機で再登録されたジョブ実行を中断されたジョブステップから続行するというジョブスケジューリング方式を特徴とする。

【0006】

【発明の実施の形態】 以下、本発明の一実施形態について図面を用いて説明する。

【0007】 図1は、本実施形態の複数の計算機から構成される汎用計算機システムの構成図であり、1つ以上

の複数の計算機により構成されるシステムを対象とする。複数の計算機である計算機 2・・3 は相互に通信を行う手段として、各計算機から更新・参照可能である領域を保持する計算機結合／負荷分散コントロール装置 1 により接続されているものとする。ジョブは計算機 2・・3 のどの計算機からも入力できる。

【0008】計算機結合／負荷分散コントロール装置 1 はジョブ分散制御部 12、資源／負荷情報収集部 13、再登録ジョブ分散制御部 14 から構成されている。ジョブは計算機 2・・3 のどの計算機からも入力でき、入力されたジョブは一旦、ジョブ分散制御部 12 へ登録され、ジョブ分散制御部 12 で選択されたもっとも最適な計算機へスケジュールされる。選択された計算機ではジョブの実行を開始する。ジョブが実行され、実行に伴い出力された実行結果、ユーザデータ及びジョブで使用する作業用領域はユーザデータ格納装置 4 へ出力する。また、ジョブの実行過程において履歴を採取したものを格納するジョブ実行履歴格納装置 5 がある。

【0009】このユーザデータ格納装置 4、ジョブ実行履歴格納装置 5 は共用装置であり、計算機 2・・3 のどの計算機からも参照、更新が可能である。計算機 2・・3 にはそれぞれ、ジョブステップ終了後、後続のジョブステップを実行すべきかどうかを判断するジョブステップ実行判定制御部 21・・31 がある。ジョブステップが終了した時点で計算機結合／負荷分散コントロール装置 1 の資源／負荷情報収集部 13 より得た情報をもとに他の計算機で実行したほうが良いと判断した場合、ジョブ再登録制御部 22・・32 で計算機結合／負荷分散コントロール装置 1 の再登録ジョブ分散制御部 14 へ再登録する。再登録負荷分散制御部 14 では最も最適な計算機を再度選択しなおし、ジョブを再登録する。再登録された計算機では再登録ジョブ実行制御部 23・・33 でジョブを受け取りジョブ実行履歴格納装置 5 より履歴データを得て、中断されていたジョブステップから実行を開始させる。

【0010】図 2 はジョブステップ実行判定部 21・・31、およびジョブステップ再登録部 22・・32 の処理の流れを示すフローチャートである。ジョブが入力される（ステップ 601）とジョブ実行の履歴をジョブ実行履歴格納装置 5 に採取し始める。ジョブステップが実行開始される（ステップ 602）と、この実行に伴い使用する入力データ、作業領域やジョブステップの実行結果をユーザデータ格納装置 4 へ出力する。ジョブステップの実行が終了する（ステップ 603）と次のステップがあるかどうかを判定（ステップ 604）し、後続のジョブステップがない場合はジョブを終了させる（ステップ 605）。後続のジョブステップがある場合（ステップ 604 YES）は資源／負荷情報収集制御部 13 より計算機 2・・3 の全ての資源／負荷情報を取得（ステップ 606）する。取得した情報から自計算機の CPU 使用

率を取出し、あらかじめ決めていた自計算機の CPU 使用率の限度を示す閾値とを比較（ステップ 607）し、閾値を下回っていた場合（ステップ 607 YES）は後続のジョブステップを実行する（ステップ 608）。

【0011】閾値を上回っていた場合（ステップ 607 YES）は負荷情報をもとに最適な計算機を選択（ステップ 609）する。選択した結果、他の計算機の方が負荷が高く、やはり自計算機と決定されることもあるので、その計算機が自計算機かどうかを判定（ステップ 610）する。自計算機の場合（ステップ 610 YES）、そのまま自計算機で後続のジョブステップを実行（ステップ 608）する。他の計算機だった場合（ステップ 610 NO）、再登録ジョブという旨の情報（フラグ）をセット（ステップ 611）し、再登録ジョブ分散制御部へジョブの再登録を行う（ステップ 612）。

【0012】図 3 は計算機結合／負荷分散コントロール装置 1 の再登録ジョブ分散制御部 14 の処理の流れを示すフローチャートである。計算機 2・・3 のジョブ再登録制御部 22・・32 よりジョブの再登録要求が行われ、その要求を受け付ける（ステップ 71）と、再登録用の専用実行待ちキューへそのジョブを登録する（ステップ 72）。その後、専用実行待ちキューからジョブの取り出しが行われる（ステップ 73）。この時点で再度、資源／負荷情報収集制御部 13 より計算機 2・・3 の全ての資源／負荷情報を取得（ステップ 74）し、取得した情報を元に最適な計算機を選定（ステップ 75）する。そして、選定された計算機へジョブの登録を行う（ステップ 76）。

【0013】図 4 は再登録ジョブ実行制御部 23・・33 の処理の流れを示すフローチャートである。再登録ジョブ分散制御部 14 よりジョブが投入される（ステップ 81）と、投入されたジョブが他の計算機で実行している再登録されたジョブかどうかを判定（ステップ 82）し、再登録されたジョブでなければ（ステップ 82 NO）、新規のジョブとしてジョブステップを実行する（ステップ 85）。他の計算機で実行していたジョブの場合（ステップ 82 YES）、当該ジョブのジョブ実行履歴を履歴格納装置 5 から読み込み（ステップ 83）、中断したジョブステップの実行のための情報回復を行い（ステップ 84）。実行すべきステップからジョブステップを再実行させる（ステップ 85）。このジョブ実行は新規に投入したどのジョブよりも最も最優先で実行される。

【0014】

【発明の効果】本発明によれば、各計算機の資源／負荷情報によってジョブステップを実行させる計算機を決定するので、ジョブ全体の実行のための負荷分散がスムーズに行え、システム全体のスループットの低下を防ぐ事ができる。

【図面の簡単な説明】

【図 1】実施形態の複数の計算機からなる汎用計算機システムの構成図である。

【図 2】実施形態のジョブステップ実行判定制御部 21・・・31、およびジョブ再登録制御部 22・・・32 の処理の流れを示すフローチャートである。

【図 3】実施形態の再登録ジョブ分散制御部 14 の処理の流れを示すフローチャートである。

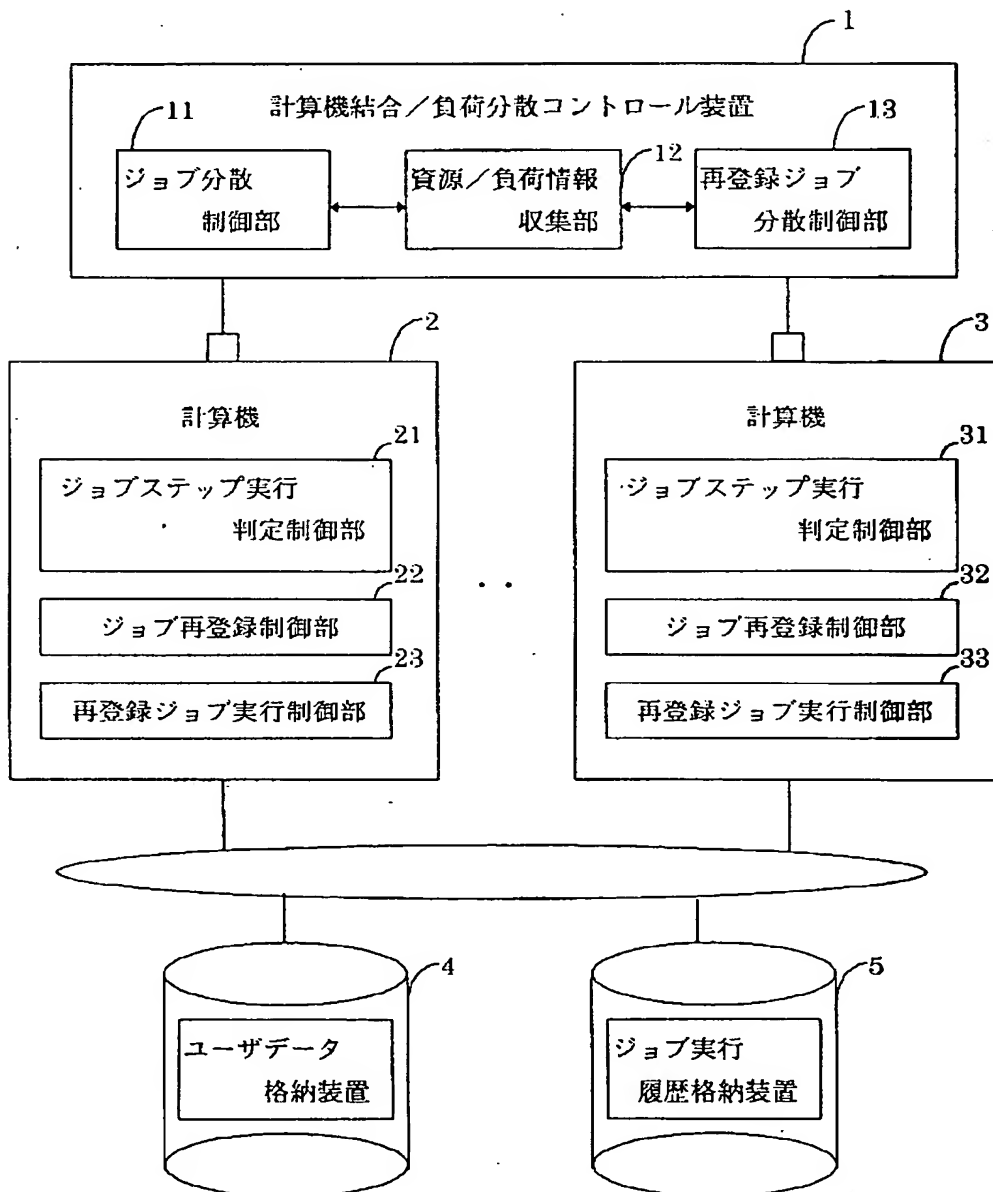
【図 4】実施形態の再登録ジョブ実行制御部 23・・・33 の処理の流れを示すフローチャートである。

【符号の説明】

1…計算機結合／負荷分散コントロール装置、11…ジョブ分散制御部、12…資源／負荷情報収集部、13…再登録ジョブ分散制御部、2・3…計算機、21・31…ジョブステップ実行判定処理部、22・32…ジョブ再登録制御部、23・33…再登録ジョブ実行制御部、4…ユーザデータ格納装置、5…ジョブ実行履歴格納装置。

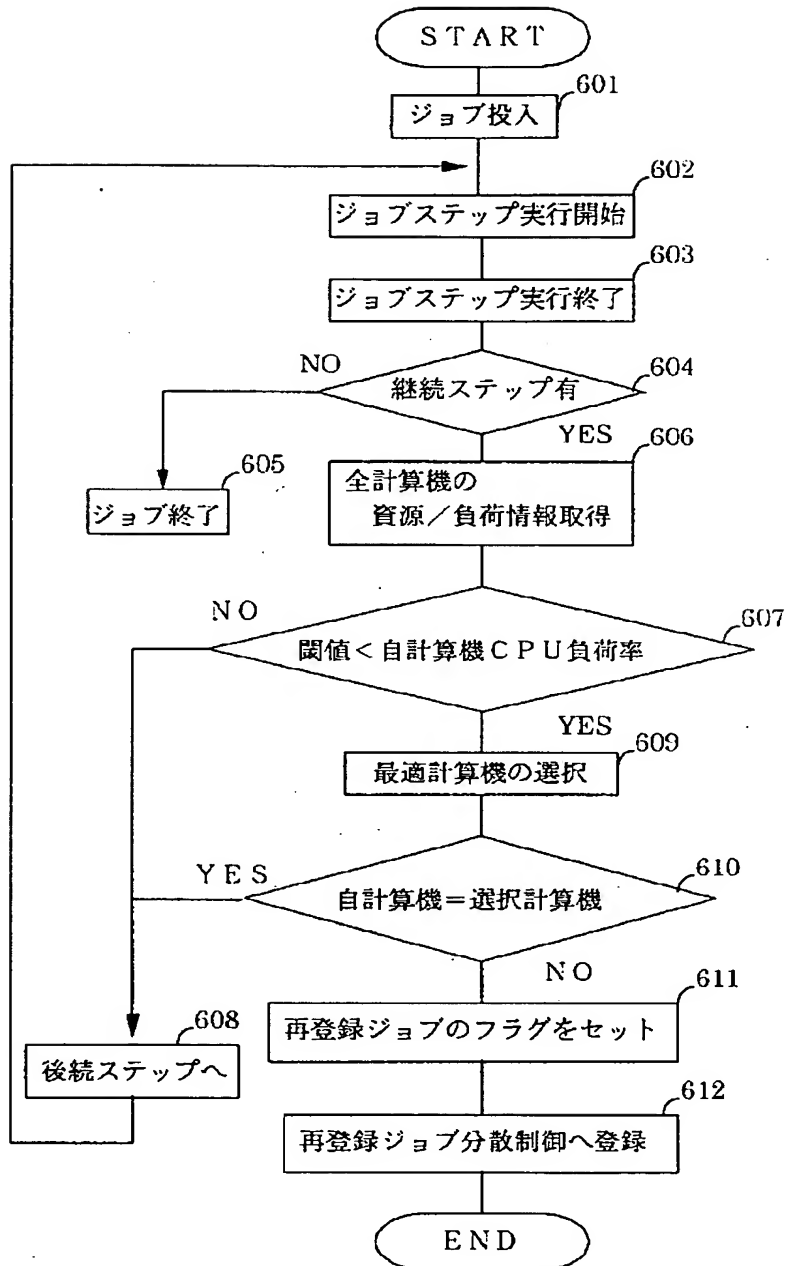
【図 1】

図 1



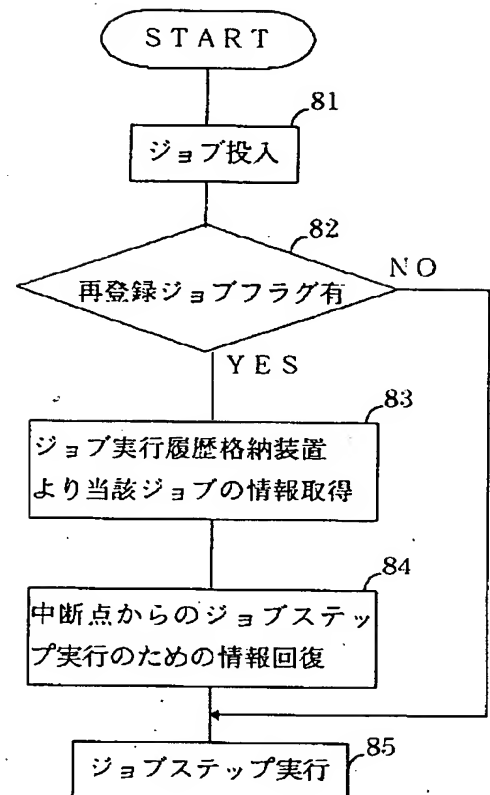
【図2】

図 2



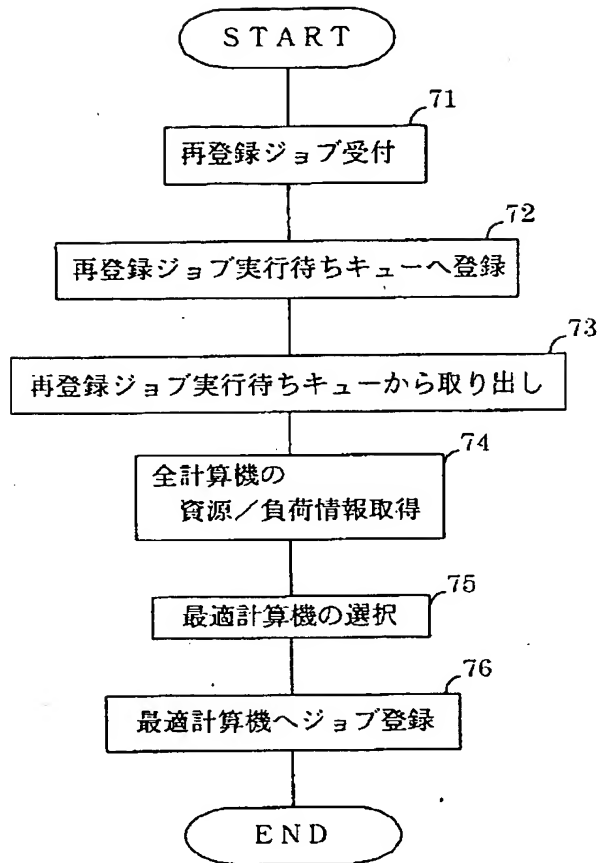
【図4】

図 4



【図3】

図 3



フロントページの続き

(72)発明者 木原 均
 神奈川県横浜市戸塚区戸塚町5030番地 株
 式会社日立製作所ソフトウェア事業部内
 (72)発明者 依知川 正利
 神奈川県横浜市戸塚区戸塚町5030番地 株
 式会社日立製作所ソフトウェア事業部内

(72)発明者 越智 直子
 神奈川県横浜市戸塚区戸塚町5030番地 株
 式会社日立製作所ソフトウェア事業部内
 Fターム(参考) 5B045 GG02 JJ08
 5B085 AC03 BA01
 5B098 AA10 GA03 GA08 GB13 GD02
 GD14